



## Tutoriais

### Comandos Loops

por: Ulisses Telemaco Neto  
[ulisses@jspbrasil.com.br](mailto:ulisses@jspbrasil.com.br)

Essa lição trata as rotinas loop do Java que poderão ser usadas em páginas JSP. Três comandos serão tratados a seguir:

```
for;
while;
do .. while;
```

#### Loop For

O loop for, repete uma instrução ou um bloco de instruções algum número de vezes até que a condição seja satisfeita. Loops for são frequentemente usados para simples iterações na qual você repete um bloco de instruções um certo número de vezes e então pára; mas você pode usar loops for para qualquer espécie de loop. O loop for no Java se parece com o seguinte:

```
for (initialization ; test ; incremet) {
statements
}
```

Onde:

initialization - é uma expressão que inicia o começo do loop;  
test - é o teste que ocorre depois de cada passo do loop;  
increment - é qualquer chamada de expressão ou função.

O código abaixo mostra um simples exemplo da utilização do laço for:

Exemplo 01:

```
<html>
<body>

<% for (int i = 2; i < 8 ; i++) { %>
<font size=<%= i %>>Bom Dia!!! - Fonte: <%= i %></font>
<% } %>

</body>
</html>
```

#### Loop While

O loop while é usado para repetir uma instrução ou bloco de instruções até que uma condição particular seja verdadeira. Loops while se parecem com o seguinte:

```
while ( condition) {
bodyOfLoop
}
```

Veja o código abaixo:

Exemplo 2:

```
<html>
<body>

<%
int i = 0;
while ( i < 8) { %>
<font size=<%= i %>>Bom Dia!!! - Fonte: <%= i %></font><br>
<%
i++;
} %>

</body>
</html>
```

Observe que se a condição é inicialmente falsa na primeira vez que é testada o corpo do loop while nunca será executado. Se você precisa executar o corpo, pelo menos uma vez, você pode usar o código do..while que será visto a seguir.

## Loop Do...While

O loop é exatamente como o loop while, exceto pelo fato de o loop do executar uma dada instrução ou bloco até que uma condição seja falsa. A diferença principal entre os dois é que os loops while testam a condição antes de iniciar o loop, tornando possível que o corpo do loop nunca seja executado caso a condição seja inicialmente falsa. Os loops do...while executam pelo menos uma vez antes de testar a condição.

Os loops do...while tem o seguinte formato:

```
do {
bodyOfLoop
}
while ( condition );
```

Veja o exemplo abaixo:

Exemplo 3:

```
<html>
<body>

<%
int i = 0;
do { %>
<font size=<%= i %>>Bom Dia!!! - Fonte: <%= i %></font><br>
<%
i++;
}
while ( i < 8 ); %>

</body>
</html>
```

## Interrupção de Loops

Em todos os loops (for, while e do), o loop termina quando a condição que você está testando é atingida. Porém, em alguma situação, você desejará sair do loop antes do seu término normal. Para isso, você pode usar as palavras chaves break e continue. A palavra chave break pára imediatamente a execução do loop corrente. Se você tiver loops aninhados dentro de loop mais externo; caso contrário, o programa simplesmente continua a execução da próxima instrução após o loop.

```
<html>
<body>

<% int j;
for (int i = 1; i < 8 ; i++) {
j = i%5;
if (j == 0){ break; }
%>
<font size=<%= i %>>Bom Dia!!! - Fonte: <%= i %></font><br>
<% } %>

</body>
</html>
```

No exemplo acima, o código dentro do loop é executado até que a variável i assuma um valor que seja divisível por 5. Quando isso ocorre, o loop é finalizado e o programa continua a ser executado após o final do código do loop. Nesse caso o resultado obtido foi uma a geração de quatro linhas de código html (ao invés de 7). Após gerar a página, o servidor JSP enviará para o navegador o seguinte código:

```

<html>
<body>

<font size=1>Bom Dia!!! - Fonte: 1</font><br>
<font size=2>Bom Dia!!! - Fonte: 2</font><br>
<font size=3>Bom Dia!!! - Fonte: 3</font><br>
<font size=4>Bom Dia!!! - Fonte: 4</font><br>

</body>
</html>

```

A palavra chave continue pára o loop e começa novamente na próxima iteração. continue é útil quando você quer ter situações de casos especiais dentro de loop.

```

<html>
<body>

<% int j;
for (int i = 1; i < 8 ; i++) {
j = i%5;
if (j == 0){ continue; }
%>
<font size=<%= i %>>Bom Dia!!! - Fonte: <%= i %></font><br>
<% } %>

</body>
</html>

```

Esse código é semelhante ao anterior, contudo, ao invés de finalizar a execução do loop, ele o reinicia em uma nova iteração. O resultado obtido é a geração de 6 linhas de código html (a linha cujo teste do if foi verdadeiro, ou seja, onde a variável i assumiu um valor divisível por 5, não foi gerada). O servidor JSP irá enviar para o navegador o seguinte código:

```

<html>
<body>

<font size=1>Bom Dia!!! - Fonte: 1</font><br>
<font size=2>Bom Dia!!! - Fonte: 2</font><br>
<font size=3>Bom Dia!!! - Fonte: 3</font><br>
<font size=4>Bom Dia!!! - Fonte: 4</font><br>
<font size=6>Bom Dia!!! - Fonte: 6</font><br>
<font size=7>Bom Dia!!! - Fonte: 7</font><br>

</body>
</html>

```